UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,262	02/27/2004	Martin Deitch	CHA920030034US1	9375

23550 7590 01/29/2008
HOFFMAN WARNICK & D'ALESSANDRO, LLC
75 STATE STREET
14TH FLOOR
ALBANY, NY 12207

EXAMINER	
PATEL, SHAMBHAVI K	

ART UNIT	PAPER NUMBER
2128	

NOTIFICATION DATE	DELIVERY MODE
01/29/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PTOCommunications@hwdpatents.com

Office Action Summary	**Application No.** 10/789,262	**Applicant(s)** DEITCH, MARTIN	
	Examiner Shambhavi Patel	**Art Unit** 2128	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 June 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-7,9-12 and 14-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 2, 4-7, 9-12 and 14-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. A **request for continued examination** under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on **14 June 2007** has been entered.

2. Claims 1, 2, 4-7, 9-12 and 14-22 have been presented for examination. Claims 3, 8 and 13 have been cancelled.

Response to Arguments

3. Applicant's arguments with respect to the 35 U.S.C. 102 and 103 rejections of claims 1, 2, 4-7, 9-12 and 14-22 have been considered but are moot in view of the new ground(s) of rejection.

4. Applicant's arguments with respect to the 35 U.S.C. 101 rejection of claims 1, 2, 4-7, 9-12 and 14-22 have been fully considered but they are not persuasive.

Regarding the 35 U.S.C. 101 rejection:

i. **Applicant submits,** on page 13 of the remarks, that outputting the behavior of the modeling overcomes the 35 U.S.C. 101 rejection.

Examiner notes that the specification does not define (or even recite) "outputting" the behavior of the modeling. Since the term in indefinite, it does not necessarily produce a real-world result. The rejection is maintained.

Specification

5. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: the specification does not define the term "computer readable medium"

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. **Claims 1, 2, 4-7, 9-12 and 14-22 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite** for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Regarding independent **claims 1, 7, 12, 16, 18-20,** the specification does not define or recite "outputting" the behavior of modeling or "outputting" the modeled workload performance. Regarding independent **claim 1,** the limitation "beginning a next modeling interval" is indefinite. Is there a first modeling interval? Regarding independent **claims 1, 7, 12 and 20,** the term "time slice dispatch mode" is indefinite. Regarding independent **claims 16, 18 and 19,** the terms "defined consumption" and "observed consumption", and the phrases "an observed consumption that does not agree with the defined consumption", "adjusting the defined consumption of each model based on the observed consumption feedback" and "the observed consumption agrees with the defined consumption" are indefinite. What is meant by "agree"? What is meant by "based on"? Regarding independent **claims 1, 7, 12 and 20,** the limitation "modeling the behavior of an LPAR" is indefinite. What is meant by "behavior"? All other claims are based by virtue of their dependency.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title..

7. **Claims 1, 2, 4-7, 9-12 and 14-22 are rejected under 35 U.S.C. 101** because the claimed invention is directed to non-statutory subject matter.

i. The Examiner asserts that the current state of the claim language is such that a reasonable interpretation of the claims would not result in any useful, concrete or tangible product. Regarding independent **claims 1, 7, 12, 16, 18-20** the specification does not define (or even recite) "outputting" the behavior of the modeling or "outputting" the modeled workload performance. Since the term in indefinite, it does not necessarily produce a real-world result. In **claim 1,** if the CP percentage is greater than the time slice percentage, no CPs are dispatched--there is no real-world result. All other claims are rejected by virtue of their dependency.

ii. **Claims 7, 18 and 20** are system claims, but they appear to claim only software elements.

iii. **Claims 12 and 19** recite "a computer readable medium". As noted above, the term "computer readable medium" is not defined in the specification. Thus, the claims fail to be limited to embodiments which fall within a statutory category

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claim(s) 1, 2, 4-7, 9-12, 14 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Rooney ('Intelligent Resource Director', 2002)** in view of **Kyne ('z/OS Intelligent Resource Director', 2001)** in view of **Buttlar ("z/CECSIM: An Efficient and Comprehensive Microcode Simulator for the IBM eServer z900"** 2002).

Regarding claim 1:

Rooney discloses a method for controlling a behavior of an LPAR (logical partition) in a computer operating in a time slice dispatch mode, comprising:

a. beginning a next time interval **(page 572 'State Sampling')**

b. calculating a resource percentage representing a percentage of total resources allocated to the LPAR **(page 573 'Maximum Processor Demand' paragraph 2:** *maximum_demand_percentage***)**

c. calculating a time slice percentage for the LPAR based on the resource percentage **(page 573 paragraph 1:** *processor_using_samples***)**

d. determining a CP (central processor) percentage representing a percentage of time that all physical CPs in the computer being modeled have been allocated to the LPAR **(page 572 'State**

Sampling'). Four times a second, every work unit in the system is sampled, to learn *where each service class is spending its time and how much each class is using each resource.*

Rooney does not explicitly disclose comparing the CP percentage and the slice percentage to then accordingly allocate resources. However, a skilled artisan would knowingly have included this functionality because if the required resources *(slice percentage)* are greater than the available resources *(CP percentage)*, then the CPs cannot be allocated to the LPAR.

Rooney does not explicitly disclose setting the resource percentage equal to 100% - a percentage of resources allocated to all other LPARs running in the simulated computer. **Kyne teaches** dividing the total amount of resources available (100%) among the LPARs running on the system **(Kyne: page 61 table at bottom of page).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney and Kyne because the method taught by Kyne provides the ability to drive a processor at 100% while providing acceptable response times for the critical applications, and ensures that all resources are utilized by the right workloads **(Kyne: page 4 1st paragraph).**

Rooney does not explicitly disclose simulating the method above. **Buttlar teaches** the verification of LPAR management software in a simulation environment **(Buttlar: paragraph 1).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney, Kyne and Buttlar because tight development schedules and a very limited supply of expensive engineering hardware make this type of verification desirable **(Buttlar 1st paragraph).**

Regarding claim 2:

Rooney discloses the method of claim 1, including the further step of repeating each of the recited steps for a next modeling interval. **(page 572 'State Sampling').** Four times a second, every work unit in the system is sampled, to learn *where each service class is spending its time and how much each class is using each resource.* Thus, the above process is repeated.

Regarding claim 4:

The combination of Rooney, Kyne and Buttlar teaches basing the percentage of resources allocated to other LPARs on a weighting factor specified for each LPAR **(Kyne: page 26 3rd paragraph),** a number of logical CPs allocated to each LPAR **(Kyne: page 48 4th paragraph)**, and a MIPS value for each LPAR **(Kyne: page 61 table at bottom of page).**

Regarding claim 5:

The combination of Rooney, Kyne and Buttlar teaches the method of claim 4, wherein the MIPS value represents a maximum consumption that each LPAR could consume in an unrestrained processor **(Kyne: page 65 2nd paragraph).**

Regarding claim 6:

The combination of Rooney, Kyne and Buttlar teaches calculating the time slice percentage through the preceding equation **(Kyne: page 55).**

Regarding claim 7:

Rooney discloses a tool for controlling operation of a computer having a system for modeling a behavior of an LPAR operating in a time slice dispatch mode, the modeling system comprising:

a. a system for calculating a resource percentage representing a percentage of total resources allocated to the LPAR **(page 573 'Maximum Processor Demand' paragraph 2:** *maximum_demand_percentage***)**

b. a system for calculating a time slice percentage for the LPAR based on the resource percentage **(page 573 paragraph 1:** *processor_using_samples***)**

c. a system for determining a CP (central processor) percentage representing a percentage of time that all physical CPs in the computer being modeled have been allocated to the LPAR **(page 572 'State Sampling').** Four times a second, every work unit in the system is sampled, to learn *where each service class is spending its time and how much each class is using each resource.*

Rooney does not explicitly disclose comparing the CP percentage and the slice percentage to then accordingly allocate resources. However, a skilled artisan would knowingly have included this functionality because if the required resources *(slice percentage)* are greater than the available resources *(CP percentage)*, then the CPs cannot be allocated to the LPAR.

Rooney does not explicitly disclose setting the resource percentage equal to 100% - a percentage of resources allocated to all other LPARs running in the simulated computer. **Kyne teaches** dividing the total amount of resources available (100%) among the LPARs running on the system **(Kyne: page 61 table at bottom of page).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney and Kyne because the method taught by Kyne provides the ability to drive a processor at 100% while providing acceptable response times for the critical applications, and ensures that all resources are utilized by the right workloads **(Kyne: page 4 1st paragraph).**

Rooney does not explicitly disclose simulating the method above. **Buttlar teaches** the verification of LPAR management software in a simulation environment **(Buttlar: paragraph 1).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney, Kyne and Buttlar because tight development schedules and a very limited supply of expensive engineering hardware make this type of verification desirable **(Buttlar 1st paragraph).**

Regarding claim 9:

The combination of Rooney, Kyne and Buttlar teaches basing the percentage of resources allocated to other LPARs on a weighting factor specified for each LPAR **(Kyne: page 26 3rd paragraph),** a number of logical CPs allocated to each LPAR **(Kyne: page 48 4th paragraph),** and a MIPS value for each LPAR **(Kyne: page 61 table at bottom of page).**

Regarding claim 10:

The combination of Rooney, Kyne and Buttlar teaches the tool of claim 9, wherein the MIPS value represents a maximum consumption that each LPAR could consume in an unrestrained processor **(Kyne: page 65 2nd paragraph).**

Regarding claim 11:

The combination of Rooney, Kyne and Buttlar teaches dividing the total amount of resources available (100%) among the LPARs running on the system **(Kyne: page 61 table at bottom of page).**

Regarding claim 12:

Rooney discloses a program product stored on a recordable medium for controlling a behavior of an LPAR in a computer operating in a time slice dispatch mode, comprising:

a. means for calculating a resource percentage representing a percentage of total resources allocated to the LPAR **(page 573 'Maximum Processor Demand' paragraph 2:** *maximum_demand_percentage***)**

b. means for calculating a time slice percentage for the LPAR based on the resource percentage **(page 573 paragraph 1:** *processor_using_samples***)**

c. means for determining a CP (central processor) percentage representing a percentage of time that all physical CPs in the computer being modeled have been allocated to the LPAR **(page 572 'State Sampling').** Four times a second, every work unit in the system is sampled, to learn *where each service class is spending its time and how much each class is using each resource.*

Rooney does not explicitly disclose comparing the CP percentage and the slice percentage to then accordingly allocate resources. However, a skilled artisan would knowingly have included this functionality because if the required resources *(slice percentage)* are greater than the available resources *(CP percentage)*, then the CPs cannot be allocated to the LPAR.

Rooney does not explicitly disclose setting the resource percentage equal to 100% - a percentage of resources allocated to all other LPARs running in the simulated computer. **Kyne teaches** dividing the total amount of resources available (100%) among the LPARs running on the system **(Kyne: page 61 table at bottom of page).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney and Kyne because the method taught by Kyne provides the ability to drive a processor at 100% while

providing acceptable response times for the critical applications, and ensures that all resources are utilized by the right workloads **(Kyne: page 4 1st paragraph).**

Rooney does not explicitly disclose simulating the method above. **Buttlar teaches** the verification of LPAR management software in a simulation environment **(Buttlar: paragraph 1).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney, Kyne and Buttlar because tight development schedules and a very limited supply of expensive engineering hardware make this type of verification deisrable **(ButtlarL 1st paragraph).**

Regarding claim 14:

The combination of Rooney, Kyne and Buttlar teaches basing the percentage of resources allocated to other LPARs on a weighting factor specified for each LPAR **(Kyne: page 26 3rd paragraph),** a number of logical CPs allocated to each LPAR **(Kyne: page 48 4th paragraph),** and a MIPS value for each LPAR **(page 61 table at bottom of page).**

Regarding claim 15:

The combination of Rooney, Kyne and Buttlar teaches dividing the total amount of resources available (100%) among the LPARs running on the system **(Kyne: page 61 table at bottom of page).**

9. **Claims 16-22 are rejected under 35 U.S.C. 103(a)** as being unpatentable over **Rooney ('Intelligent Resource Director', 2002) in view of Buttlar ("z/CECSIM: An Efficient and Comprehensive Microcode Simulator for the IBM eServer z900"** 2002).

Regarding claim 16:

Rooney discloses a method for tracking workload performance of a plurality of LPARs in a computer, comprising

a. providing each LPAR specified in the computer, wherein each LPAR includes a defined consumption that is dependent on a consumption of the other LPARs **(page 575 2nd paragraph)**

b. setting an initial defined consumption for each LPAR, and running each LPAR and determining an observed consumption for each LPAR **(page 571 'WLM CPU weight-management configuration' 2nd paragraph** *initial weight* and *current weight***)**

c. comparing the observed consumption with the defined consumption for all of the LPAR **(page 575 'Receiver Processing' 1st paragraph).** The LPAR weight fix routine determines whether the receiver CPU delay bottleneck can be addressed by raising the partition processor weight *(consumption)*.

d. for each LPAR that has an observed consumption that does not agree with the defined consumption, feeding the observed consumption back to the other LPAR **(page 575 'Donor Selection').** After it has been determined that the weight *(consumption)* of a service class needs to be increased, a donor whose weight *(consumption)* must be reduced as a result of this increase is selected.

e. adjusting the defined consumption of each LPAR based on the feedback **(page 576 'Donor Projections' 1st paragraph).** If a good trade is found, the partition weights of the receiver and donor are adjusted.

f. iteratively repeating the running, comparing, feeding and adjusting steps until the observed consumption agrees with the defined consumption for each LPAR. **(page 572 'Policy-adjustment framework').** This is repeated every ten seconds for every receiver class in need of resource allocation.

Rooney does not explicitly disclose simulating and modeling the method above. **Buttlar teaches** the verification of LPAR management software in a simulation environment **(Buttlar: paragraph 1).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney and Buttlar because tight development schedules and a very limited supply of expensive engineering hardware make this type of verification desirable **(Buttlar 1st paragraph).**

Regarding claim 17:

Rooney discloses the method of claim 16, wherein the consumption is a measure of processor resources consumed by each LPAR **(page 571 'WLM CPU weight-management configuration' 2nd paragraph).**

Regarding claim 18:

Rooney discloses a computer tool for tracking workload performance of a plurality of LPARs in a computer, comprising

a. a system for building each LPAR specified in the computer, wherein each LPAR includes a defined consumption that is dependent on a consumption of the other LPARs **(page 575 2nd paragraph)**

b. a system for running each LPAR and determining an observed consumption for each model **(page 571 'WLM CPU weight-management configuration' 2nd paragraph** *initial weight* and *current weight***)**

c. a system for comparing the observed consumption with the defined consumption for all of the LPAR **(page 575 'Receiver Processing' 1st paragraph).** The LPAR weight fix routine determines whether the receiver CPU delay bottleneck can be addressed by raising the partition processor weight *(consumption).*

d. a system for feeding back the observed consumption to the other models from each LPAR that has an observed consumption that does not agree with the defined consumption **(page 575 'Donor Selection').** After it has been determined that the weight *(consumption)* of a service class needs to be increased, a donor whose weight *(consumption)* must be reduced as a result of this increase is selected.

e. a system for adjusting the defined consumption of each LPAR based on the feedback **(page 576 'Donor Projections' 1st paragraph).** If a good trade is found, the partition weights of the receiver and donor are adjusted.

f. a system for iteratively repeating the running, comparing, feeding and adjusting steps until the observed consumption agrees with the defined consumption for each LPAR. **(page 572 'Policy-**

adjustment framework'). This is repeated every ten seconds for every receiver class in need of resource allocation.

Rooney does not explicitly disclose simulating the method above. **Buttlar teaches** the verification of LPAR management software in a simulation environment **(Buttlar: paragraph 1).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney and Buttlar because tight development schedules and a very limited supply of expensive engineering hardware make this type of verification desirable **(Buttlar 1st paragraph).**

Regarding claim 19:

Rooney discloses a program product stored on a recordable medium for tracking workload performance of a plurality of LPARs in a computer, comprising

a. means each LPAR specified in the computer, wherein each LPAR includes a defined consumption that is dependent on a consumption of the other LPARs **(page 575 2nd paragraph)**

b. means for running each model and determining an observed consumption for each LPAR **(page 571 'WLM CPU weight-management configuration' 2nd paragraph** *initial weight* and *current weight*)

c. means for comparing the observed consumption with the defined consumption for all of the LPAR **(page 575 'Receiver Processing' 1st paragraph).** The LPAR weight fix routine determines whether the receiver CPU delay bottleneck can be addressed by raising the partition processor weight *(consumption)*.

d. means for feeding back the observed consumption to the other LPAR from each LPAR that has an observed consumption that does not agree with the defined consumption **(page 575 'Donor Selection').** After it has been determined that the weight *(consumption)* of a service class needs to be increased, a donor whose weight *(consumption)* must be reduced as a result of this increase is selected.

e. means for adjusting the defined consumption of each LPAR based on the feedback **(page 576 'Donor Projections' 1st paragraph).** If a good trade is found, the partition weights of the receiver and donor are adjusted

f. means for iteratively repeating the running, comparing, feeding and adjusting steps until the observed consumption agrees with the defined consumption for each LPAR. **(page 572 'Policy-adjustment framework').** This is repeated every ten seconds for every receiver class in need of resource allocation.

Rooney does not explicitly disclose simulating the method above. **Buttlar teaches** the verification of LPAR management software in a simulation environment **(Buttlar: paragraph 1).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney and Buttlar because tight development schedules and a very limited supply of expensive engineering hardware make this type of verification desirable **(Buttlar 1st paragraph).**

Regarding claims 20-22:

Rooney discloses a computer tool for controlling LPAR behavior comprising:

a. means for calculating a resource percentage representing a percentage of total resources allocated to the LPAR **(page 573 'Maximum Processor Demand' paragraph 2:** *maximum_demand_percentage***)**

b. means for calculating a time slice percentage for the LPAR based on the resource percentage **(page 573 paragraph 1:** *processor_using_samples***)**

c. means for determining a CP (central processor) percentage representing a percentage of time that all physical CPs in the computer being modeled have been allocated to the LPAR **(page 572 'State Sampling').** Four times a second, every work unit in the system is sampled, to learn *where each service class is spending its time and how much each class is using each resource.*

d. means for building each LPAR specified in the computer simulation, wherein each LPAR includes a defined consumption that is dependent on a consumption of the other LPARs **(page 575 2nd paragraph)**

e. means for running each LPAR and determining an observed consumption for each LPAR **(page 571 'WLM CPU weight-management configuration' 2nd paragraph** *initial weight* and *current weight*)

f. means for comparing the observed consumption with the defined consumption for all of the LPAR **(page 575 'Receiver Processing' 1st paragraph).** The LPAR weight fix routine determines whether the receiver CPU delay bottleneck can be addressed by raising the partition processor weight *(consumption)*.

g. means for feeding back the observed consumption to the other models from each LPAR that has an observed consumption that does not agree with the defined consumption **(page 575 'Donor Selection').** After it has been determined that the weight *(consumption)* of a service class needs to be increased, a donor whose weight *(consumption)* must be reduced as a result of this increase is selected.

h. means for adjusting the defined consumption of each LPAR based on the feedback **(page 576 'Donor Projections' 1st paragraph).** If a good trade is found, the partition weights of the receiver and donor are adjusted

i. means for iteratively repeating the running, comparing, feeding and adjusting steps until the observed consumption agrees with the defined consumption for each LPAR. **(page 572 'Policy-adjustment framework').** This is repeated every ten seconds for every receiver class in need of resource allocation.

Rooney does not explicitly disclose comparing the CP percentage and the slice percentage to then accordingly allocate resources. However, a skilled artisan would knowingly have included this functionality because if the required resources *(slice percentage)* are greater than the available resources *(CP percentage)*, then the CPs cannot be allocated to the LPAR.

Rooney does not explicitly disclose simulating the method above. **Buttlar teaches** the verification of LPAR management software in a simulation environment **(Buttlar: paragraph 1).** At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Rooney and Buttlar because

tight development schedules and a very limited supply of expensive engineering hardware make this type of verification desirable **(Buttlar 1st paragraph).**

Conclusion

10. **Examiner's Note:** Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. In the case of amending the claimed invention, Applicant is respectfully requested to indicate the portion(s) of the specification which dictate(s) the structure relied on for proper interpretation and also to verify and ascertain the metes and bounds of the claimed invention

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shambhavi Patel whose telephone number is (571) 272-5877. The examiner can normally be reached on Monday-Friday, 8:00 am – 4:30 pm.

11. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on (571) 272-2279. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

SKP

HUGH JONES Ph.D.
PRIMARY PATENT EXAMINER
TECHNOLOGY CENTER 2100